



⑮ BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENT- UND  
MARKENAMT

⑫ **Offenlegungsschrift**  
⑩ **DE 100 15 114 A 1**

⑤① Int. Cl.<sup>7</sup>:  
**G 06 F 17/50**  
B 62 D 65/00

⑲ Aktenzeichen: 100 15 114.0  
⑳ Anmeldetag: 28. 3. 2000  
㉑ Offenlegungstag: 4. 10. 2001

DE 100 15 114 A 1

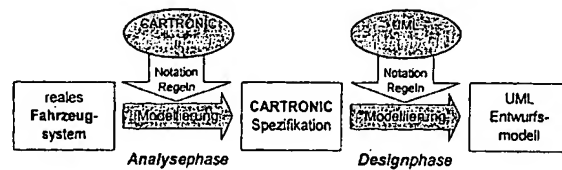
㉒ Anmelder:  
Robert Bosch GmbH, 70469 Stuttgart, DE

㉓ Erfinder:  
Flores, Pio Torre, 70191 Stuttgart, DE; Schirmer,  
Juergen, Dr., 69124 Heidelberg, DE; Walther,  
Michael, Dr., 71696 Möglingen, DE; Huelser, Holger,  
Dr., 70329 Stuttgart, DE; Bertram, Torsten, 40547  
Düsseldorf, DE; Heckes, Marc, 47199 Duisburg, DE;  
Petersen, Joerg, 45894 Gelsenkirchen, DE

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

⑤④ Verfahren und Vorrichtung zur Modellierung eines mechatronischen Systems in einem Kraftfahrzeug

⑤⑦ Es werden ein Verfahren und Vorrichtung zur Modellierung eines mechatronischen Systems in einem Kraftfahrzeug im Rahmen eines objektbasierten Ordnungskonzept als eine Abbildung in die Unified Modeling Language beschrieben. Die Elemente der CARTRONIC mit Komponenten und Hüllen als ihren Klassen beziehungsweise Objekten und Aufträgen (mit Rückmeldung), Abfragen (mit Hinweis) und Anforderungen als ihren Kommunikationsbeziehungen sind an Hand von Beispielen zusammen mit den wesentlichen Regeln aus dem Gesamtregelwerk vorgestellt. Für diese Modellierungselemente wird eine Abbildungsvorschrift in UML-Konstrukte dargestellt.



DE 100 15 114 A 1

- 5 [0001] Die Erfindung betrifft ein Verfahren und Vorrichtung zur Modellierung eines mechatronischen Systems in einem Kraftfahrzeug.
- [0002] Die Forderung nach mehr Sicherheit, Wirtschaftlichkeit, Komfort sowie einer besseren Umweltverträglichkeit lässt die Mechatronik im Fahrzeug zu einem immer bedeutenderen und wettbewerbsbestimmenden Faktor werden. Wirtschaftliche Fahrzeugentwicklungen und die Beherrschung komplexer Systemstrukturen bei immer kürzer werden-
- 10 den Produktzyklen erzwingen durchgängige, möglichst weit automatisierte, rechnerunterstützte Entwicklungsprozesse. In der Analysephase können auf der Basis vereinbarter, formaler Strukturierungs- und Modellierungsregeln des automobilhersteller- und zuliefererneutralen Ordnungskonzepts "Cartronic modulare", erweiterbare Architekturen für "Funktion", "Sicherheit" und "Elektronik" spezifiziert werden.
- [0003] Die Forderung nach mehr Sicherheit, Wirtschaftlichkeit, Komfort und einer besseren Umweltverträglichkeit lässt die Mechatronik im Fahrzeug zu einem immer bedeutenderen und wettbewerbsbestimmenden Faktor im Technologiewandel von der Mechanik über die Elektronik zur Informationstechnik werden. Bei ständig steigender Komplexität der Systeme und gleichzeitig immer kürzer werdenden Produktzyklen bleibt der Kosten- und Entwicklungsaufwand nur bei Einsatz eines durchgängigen, möglichst weit automatisierten, rechnerunterstützten sowie weitgehend parallelisierten Arbeits- und Entwicklungsprozesses beherrschbar.
- 15 [0004] Ein Ansatz zur Lösung der teilweise divergierenden Anforderungen ist die Vernetzung der bisher weitgehend unabhängig voneinander arbeitenden Einzelsysteme zu einem fahrzeugweiten Verbundsystem und die logische Zusammenfassung von Systemkomponenten zu funktionalen Einheiten mit standardisierten Schnittstellen. Der Systemverbund bietet die Möglichkeit einer Kooperation und Mehrfachnutzung von Sensorik sowie Aktuatorik und somit einer Nutzbar-
- 20 machung emergenter Funktionen.
- [0005] Die Vernetzung ermöglicht darüber hinaus einen Wandel von rein funktionsorientierten Realisierungen zu Konfigurationen, bei denen die Anwendungsfunktionen auf vernetzte Steuergeräte abgebildet werden. Außerdem können bei partiellen Systemausfällen dynamische Verlagerungen von Funktionen auf andere Systeme unterstützt werden.
- [0006] Ausgehend von den logischen Funktionseinheiten mit ihren standardisierten Schnittstellen wird es ebenfalls möglich, Funktionen unterschiedlichen Ursprungs, verschiedener Automobilhersteller und Zulieferer, miteinander zu vernetzen. Ein Funktionslieferant muss hierbei garantieren, dass die Funktion auch bei Verteilung auf mehrere vernetzte Steuergeräte die geforderte Spezifikation einhält.
- 25 [0007] Die Entwicklung komplexer, vernetzter Systeme setzt einen systematischen Prozess mit rekursiven Phasen und den Einsatz rechnergestützter Werkzeuge voraus, bei dem sowohl der Automobilhersteller als auch der Zulieferer alle Anforderungen und Randbedingungen an die zu entwickelnde Funktion formulieren, die vielfältigen Interaktionen mit anderen Funktionen und der Umgebung in allen Anwendungs- und Fehlerfällen analysieren und die Funktion in ihrer Auswirkung auf die Sicherheit des Gesamtverbunds bewerten kann. Für die Entwicklung von komplexen, vernetzten Systemen hat sich das V-Modell als Vorgehensmodell auch in der Automobilbranche etabliert. Das V-Modell sieht vor, dass sämtliche Aktivitäten und Abläufe zur Funktionsentwicklung in 11 Phasen eingeordnet werden können (Fig. 1).
- 30 [0008] Das V-Modell beschreibt ein Vorgehen, bei dem der Spezifikations- und Entwurfsprozess durch die Detaillierung und Verfeinerung charakterisiert sind und das sich als top-down-Vorgehensweise veranschaulichen lässt. Dagegen sind die Verifikations- und Validierungsphasen bottom-up-Vorgehensweisen. Wesentliche Anforderung und Voraussetzung für Qualitätszertifizierungen sind hierbei detaillierte Dokumentationsunterlagen für jede einzelne Phase.
- [0009] Um den Forderungen nach einer wirtschaftlichen Fahrzeugentwicklung, der Beherrschung komplexer Systemstrukturen und einer adäquaten Dokumentation gerecht werden zu können, wurde das Ordnungskonzept "Cartronic" (siehe Bertram, T., R. Bitzer, R. Mayer und A. Volkart, 1998, CARTRONIC – An open architecture for networking the control systems of an automobile, Detroit/Michigan, USA, SAE 98200, 1-9) entwickelt.
- 35 [0010] In der ersten Phase der Prozesskette, der Analyse, ermöglicht das auf objektbasierenden Grundgedanken entwickelte Ordnungskonzept die logische Zusammenfassung von Systemkomponenten zu funktionalen Einheiten mit standardisierten, logischen Schnittstellen. Die Beschreibung der Vernetzung der bisher weitgehend unabhängig voneinander arbeitenden Einzelsysteme eines Kraftfahrzeugs zu einem fahrzeugweiten Verbundsystem stellt somit ein (Meta-)Modell für eine modular erweiterbare Funktions-, Sicherheits- und Elektronikarchitektur dar. Ein wesentlicher Vorteil dieser automobilhersteller- und zuliefererneutralen Spezifikationsmöglichkeit ist die nach kurzer Einarbeitungszeit allen am Entwicklungsprozess Beteiligten verständliche, logische Beschreibung der Anforderungen schon zu einem sehr frühen Entwicklungszeitpunkt.
- 40 [0011] Grafikbasierte Modelle unterstützen als wesentliche Dokumentationselemente während aller Entwicklungsphasen eine Kommunikation zwischen allen an der Entwicklung beteiligten Personen sowie nach Abschluss der Entwicklung die Pflege und Weiterentwicklung. Ergänzend zum klassischen Softwareengineering sind dabei für mechatronische Systementwicklungen im Kraftfahrzeugbereich als Personengruppen zu unterstützen:
- [0012] Fahrzeughersteller als Benutzer/Kunden sowie Interessenten, die Informationen zur Funktionalität des Systems benötigen,
- 45 Ingenieure der Fachrichtungen Maschinenbau und Elektrotechnik sowie Informatiker als Entwickler der mechatronischen Komponenten auf Hersteller- und Zuliefererseite sowie diejenigen, die diese nach abgeschlossener Entwicklung modifizieren beziehungsweise erweitern werden und dazu Verständnis des Gesamtsystems oder seiner Teile benötigen, Manager auf Entwickler- und Kundenseite, die organisatorische und wirtschaftliche Einzelheiten zur Projektkontrolle, Berechnung von Kosten und Informationen für zukünftige Projekte und Entwicklungen benötigen sowie nicht zuletzt
- 50 die Fahrzeugführer als spezielle Endbenutzer, die mit ausgewählten Funktionalitäten des Systems vertraut gemacht werden müssen.
- [0013] Ein wesentlicher Schritt am Ende der Analyse- und zu Beginn der Designphase ist die Abbildung der in Cartronic

nic entwickelten Spezifikationsmodelle in einen informationstechnischen Entwurf für die Softwareentwicklung. Diese Abbildung trägt zur Erhöhung der Informationsdichte und der Erweiterung des semantischen Gehalts aufgestellter Cartronic-Modelle bei, definiert Teilsysteme in einer Gesamtsystemarchitektur, erhöht die Transparenz des Gesamtsystems in Zielrichtung dessen Implementierung und schafft wesentliche Grundlagen für eine verteilte Entwicklung und Testbarkeit von mechatronischen Systemen.

[0014] Vorliegend wird eine Abbildung von in CARTRONIC® erstellten Spezifikationsmodellen in eine standardisierte, objektorientierte Darstellung vor dem Hintergrund einer möglichst weitgehenden Unterstützung durch kommerziell verfügbare Software-Entwicklungs-Werkzeuge beschrieben. Eine dafür erforderliche, geeignete Notation stellt der von der Object Management Group (OMG) international standardisierte, objektorientierte Sprachstandard der Unified Modeling Language (UML) dar.

[0015] Im folgenden wird eine zusammenfassende Beschreibung der Strukturelemente und formalen Strukturierungs- sowie Modellierungsregeln nach CARTRONIC® für Funktionsarchitekturen gegeben. Ferner wird auf einer hohen Abstraktionsebene eine Funktionsarchitektur des Gesamtfahrzeugs mit einer Detaillierung für die Komponente Antrieb vorgestellt. Davon ausgehend erfolgt zunächst die Darstellung theoretischer Grundlagen der Modellierung, bevor auf die verwendeten Elemente der objektorientierten Notation mit UML im weiteren Verlauf dieses Abschnitts eingegangen wird. Die Vorgehensweise für das vorhandene Modell nach CARTRONIC® wird an einem Beispiel aufgezeigt.

[0016] Ein bereits in heutigen Fahrzeugen existierendes Beispiel für einen Systemverbund ist die Antriebsschlupfregelung. Diese wird erst durch die Kommunikation des Steuergeräts für die Antriebsschlupfregelung mit dem Steuergerät für das Motormanagement zur Regelung des Antriebsmoments möglich.

[0017] CARTRONIC® ist ein Ordnungskonzept für alle Steuerungs- und Regelungssysteme eines Fahrzeugs. Das Konzept enthält modulare erweiterbare Architekturen für "Funktion", "Sicherheit" und "Elektronik" auf der Basis vereinbarter formaler Strukturierungs- und Modellierungsregeln.

[0018] Unter einer Architektur ist hier sowohl die Strukturierungssystematik (Regeln) zu verstehen als auch deren Umsetzung in eine konkrete Struktur. Die Funktionsarchitektur umfasst sämtliche im Fahrzeug vorkommenden Steuerungs- und Regelungsaufgaben. Die Aufgaben des Systemverbunds werden logischen Komponenten zugeordnet, die Schnittstellen der Komponenten und ihr Zusammenwirken werden festgelegt. Die Sicherheitsarchitektur erweitert die Funktionsarchitektur um Elemente, die einen sicheren Betrieb des Systemverbunds garantieren. Schließlich wird für die Elektronik eine Systematik angegeben, wie der Systemverbund mit bedarfsgerecht optimierten Hardwaretopologien zu realisieren ist.

[0019] Die Elemente der Architekturen sind Komponenten und Kommunikationsbeziehungen auf der einen und Strukturierungs- und Modellierungsregeln auf der anderen Seite. Im Rahmen der Strukturierung wird von einem System als einer Zusammenstellung von Komponenten zu einem Ganzen gesprochen, die über Kommunikationsbeziehungen miteinander in Wechselwirkungen stehen. Der Begriff Komponente meint nicht zwangsläufig eine physikalische Einheit im Sinne eines Bauteils, sondern wird als Funktionseinheit verstanden. Bei dem Ordnungskonzept werden drei verschiedene Typen von Komponenten unterschieden:

- Komponenten mit überwiegend koordinierenden und verteilenden Aufgaben,
- Komponenten mit hauptsächlich operativen und ausführenden Aufgaben und
- Komponenten, die ausschließlich Informationen generieren und bereitstellen.

[0020] Bei den Kommunikationsbeziehungen wird zwischen einem Auftrag (mit Rückmeldung), einer Abfrage (mit Hinweis) und einer Anforderung unterschieden. Den Auftrag kennzeichnet die Pflicht zur Ausführung; für den Fall der Nichterfüllung muss der Auftragnehmer eine Rückmeldung an den Auftraggeber absetzen, die den Grund für die Nichtausführung beschreibt. Die Abfrage dient der Beschaffung von Informationen für eine Auftragsausführung. Für den Fall, dass eine Komponente die abgefragte Information nicht bereitstellen kann, gibt sie einen Hinweis an die fragende Komponente. Eine Anforderung beschreibt einen "Wunsch", dass eine Funktion von einer anderen Komponente ausgeführt wird. An die Anforderung ist allerdings nicht die Pflicht zur Erfüllung gekoppelt, was beispielsweise bei konkurrierenden Anforderungen Berücksichtigung findet. Folgende Tabelle stellt die Strukturelemente zusammenfassend dar.

STRUKTURELEMENT	KURZBESCHREIBUNG
Komponente	Logische Funktionseinheit.
Hülle	Von einer detaillierten Komponente bleibt eine Hülle, die die Kommunikationen an die Teilkomponenten weiterleitet sowie eine „ist Teil von“-Beziehung ausdrückt („Sicht von außen nach innen“).
System	Ein System besteht aus mehreren Komponenten und (Sub-) Systemen („Sicht von innen nach außen“).
Auftrag (mit Rückmeldung)	Handlungsanweisung mit Pflicht zur Ausführung einer Funktion.
Abfrage (mit Hinweis)	Ermittlung einer Information.
Anforderung	„Wunsch“ auf Ausführung einer Funktion.
Regeln	Regeln zu: <input type="checkbox"/> Kommunikationsbeziehungen <input type="checkbox"/> Modellierungsmustern

[0021] Die Strukturierungsregeln beschreiben erlaubte Kommunikationsbeziehungen innerhalb der Architektur des Gesamtfahrzeugs. Es werden Strukturierungsregeln unterschieden, die die Kommunikationsbeziehungen auf der gleichen Abstraktionsebene und in höhere und tiefere Ebenen unter Berücksichtigung angegebener Randbedingungen regeln. Ferner klären die Strukturierungsregeln die Weiterleitung von Kommunikationsbeziehungen von einem System in ein anderes in dessen Detaillierung hinein.

[0022] Die Modellierungsregeln beinhalten Muster, die Komponenten und Kommunikationsbeziehungen für die Lösung spezieller, mehrfach vorkommender Aufgaben zusammenfassen. Diese Muster, zum Beispiel ein Energiemanagement, können dann an verschiedenen Stellen innerhalb der Struktur des Fahrzeugs wiederverwendet werden.

[0023] Eine nach den Strukturierungs- und Modellierungsregeln entwickelte Struktur zeichnet sich durch folgende Merkmale aus:

- vereinbarte, einheitliche Strukturierungs- und Modellierungsregeln auf allen Abstraktionsebenen,
- hierarchische Auftragsflüsse,
- hohe Eigenverantwortung der einzelnen Komponenten,
- Bedienelemente, Sensoren und Schätzer sind gleichwertige Informationsgeber und eine
- Kapselung, die jede Komponente für die übrigen Komponenten so sichtbar wie nötig und so unsichtbar wie möglich dargestellt.

[0024] Fig. 2 stellt beispielhaft die Architekturmerkmale und die erlaubten Kommunikationsbeziehungen dar. Dies sind im Einzelnen – der Einfachheit halber wird nur von Auftrag, Abfrage und Anforderung gesprochen, gemeint ist aber die Beziehung, die diese jeweils ermöglichen:

- der Auftrag moment\_ga (Moment am Getriebeausgang bereitstellen), der von der Hülle Antrieb an die Eingangs-

komponente Antriebs-Koordinator weitergeleitet wird, die gleichzeitig auch Koordinator ist,

- die Aufträge `moment_ma` (Moment am Motorausgang bereitstellen), `stelleKraftschluss` und `einlegenGang` (einlegen eines Ganges) vom Antriebs-Koordinator an Motor, Wandler und Getriebe,
- die Anforderung `!rGang` (Rückwärtsgang) vom Getriebe-Bedienfeld an den Antriebs-Koordinator,
- die Abfragen `?zustand` und `?luftdruck` (der Umgebung) an Getriebe und Umwelt
- sowie die Abfragen `?gang` (Rückwärtsgang oder nicht) und `?drehzahl` an die Hülle Antrieb, die diese an die zuständigen Komponenten Motor und Getriebe weiterleitet.

[0025] Im klassischen Software-Lebenszyklus werden die streng sequentiell zu durchlaufenden Phasen Problemanalyse, Systemspezifikation, Entwurf, Implementierung mit Komponenten-, Gesamttest und Einführung sowie Betrieb und Pflege eines Softwaresystems unterschieden. In der Praxis ist ein solcher, streng sequentieller Entwicklungsprozess eine nicht einzuhaltende Idealisierung. Theoretisch klar abgrenzbare Punkte überlappen sich oder sind unter Umständen unterschiedlich weit vorangeschritten; gleichzeitig schreitet das Know-how auf Seiten aller am Entwicklungsprozess Beteiligten mit der Systementwicklung weiter voran. Eine objektorientierte Vorgehensweise ermöglicht ein phasenübergreifendes Vorgehen mit von Anfang an hoher Wiederverwendbarkeit bereits entwickelter beziehungsweise vorhandener Anteile und Konzepte. Dies wird durch die Verwendung einer rechnerunterstützten, grafischen Notation wesentlich erleichtert. Die in der objektorientierten Softwareentwicklung eingesetzten, verschiedenen, methodischen Vorgehensweisen beinhalten eine speziell für die jeweilige Methode entwickelte, grafische Notation. Die UML ist hervorgegangen aus den drei in der industriellen Softwareentwicklung meist verwendeten Methoden, der Booch-Methode, benannt nach Grady Booch, der unter James Rumbaugh entwickelten Object Modeling Technique (OMT) sowie dem unter Ivar Jacobson entwickelten Object Oriented Software Engineering (OOSE). Die UML stellt dabei keine weitere, neue, universelle Methode dar, sondern ein Metamodell zur Konstruktion von Modellen auf verschiedene Sichten (Fig. 3). Sie stellt eine grafische und ergänzend tabellarische und textuelle Notation mit einheitlicher Syntax und eindeutig definierter Semantik dar.

[0026] Entwickelte UML-Modelle sind von allen am Entwicklungsprozess beteiligten Personen eindeutig interpretierbar und bieten als wesentliche Vorteile:

- die Verwendung eines internationalen Standards,
- eine möglichst herstellerunabhängige, toolunterstützte Vorgehensweise,
- eine Aufweichung der starren Einhaltung der klassischen Hintereinanderabfolge von Analyse- und Designphase bei der Softwareentwicklung ohne Aufgabe des Software-Lebenszyklus-Modells als Grundlage einer ingenieurmäßigen top-down-Vorgehensweise,
- möglichst weitgehende Unabhängigkeit von der letztendlich verwendeten Programmiersprache auf der logischen Ebene,
- die Erhaltung der Konsistenz zwischen Analyse, Designentwurf und Implementierung sowie
- die Möglichkeit einer gleichzeitigen bottom-up-Reverse-Engineering-Vorgehensweise.

[0027] In der Analysephase entstehen CARTRONIC®-Modelle als eine präformale strukturierte Spezifikation, was das mechatronische System leisten soll. Diese Modelle stellen eine objektbasierte Abstraktion der funktional logischen Konzepte aus den Fahrzeugsystemstrukturen dar. Durch eine geeignete Abbildung in ein weitaus mächtigeres UML-Modell findet gleichzeitig der Wechsel von der Analyse- in die Design- und Entwurfsphase statt. Dabei werden Grundlagen für die Gesamtarchitektur des Softwaresystems gelegt, Teilsysteme zur Reduktion beziehungsweise Beherrschbarkeit der Komplexität definiert und saubere Schnittstellen zwischen diesen spezifiziert. Die Hinzufügung von mehr und mehr Details führt im fortschreitenden Entwurfsprozess in Richtung Implementierung. Das Ziel der Design- beziehungsweise Entwurfsphase besteht in der Festlegung der Systemkomponenten, deren Aufbau und Schnittstellen mit der Definition des zugrundeliegenden logischen Datenmodells einschließlich der Daten- und algorithmischen Strukturen sowie deren Validierung. Komplexität ist durch Abstraktion zu bewältigen, wobei sowohl die Einfachheit als auch Überschaubarkeit des Ganzen gewährleistet sein muss (Strukturierung im Großen). In späteren Schritten bezieht sich die Strukturierung ebenso auf die Auswahl angemessener Programmbausteine bei der algorithmischen Formulierung mit dem Ziel einer Optimierung der geforderten Leistungseigenschaften des Systems (Strukturierung im Kleinen). Das Ziel der Implementierungsphase besteht in der Übertragung des logischen Datenmodells, der Systemarchitektur und Algorithmen in übersetzbaren Programmcode für die einzelnen Steuergeräte und das Kommunikationsnetzwerk im Kraftfahrzeug.

#### Zeichnung

[0028] Die Erfindung ist anhand der in den Fig. 1 bis 14 dargestellten Ausführungsformen näher erläutert.

#### Beschreibung von Ausführungsbeispielen

[0029] Im Folgenden wird die Abbildung der Cartronic-Elemente in UML-Elemente anhand des in Fig. 4 dargestellten Ausschnittes vorgestellt. Wesentliche Überlegungen sind hierbei, die CARTRONIC®-Regeln möglichst komplett zu unterstützen, die Grundgedanken der Objektorientierung zu wahren und dabei für den CARTRONIC®-Modellierer verständlich genug und ausreichend transparent zu bleiben sowie alle notwendigen Informationen für spätere Arbeitsschritte aufnehmen und darstellen zu können.

[0030] Fig. 4 zeigt die CARTRONIC®-Komponenten Umwelt, Antrieb, Fahrzeug-Koordinator und elektrisches Bordnetz. Zwischen diesen Komponenten bestehen folgende Kommunikationsbeziehungen: Der Fahrzeug-Koordinator befragt den Antrieb nach der aktuellen `?drehzahl` beziehungsweise beauftragt den Antrieb ein `moment_ga` am Getriebeausgang bereitzustellen. Der Antrieb befragt den Informationsgeber Umwelt nach dem aktuellen `?luftdruck` und fordert

!el\_Leistung von der Komponente elektrisches Bordnetz an.

- [0031] Klassen in der objektorientierten Terminologie sind in der Regel die Generalisierungen gleichartiger Objekte (Schablonen), auf höheren Abstraktionsebenen sind Komponenten beziehungsweise Klassen seltener materielle Gegenstände, sondern meistens abstrakte Gebilde oder Funktionseinheiten. Objekte (im Allgemeinen Gegenstände) sind Exemplare von Klassen mit Eigenschaften und Verhalten. In der objektorientierten Modellierung ist ein häufig genutzter Einstieg bei der Suche nach Klassen die Suche nach Substantiven, da diese in der Sprache im Allgemeinen die Generalisierung von Objektgruppen bilden. Adjektive beschreiben Eigenschaften und werden in der Regel als Attribute modelliert. Operationen wiederum bilden das Verhalten von Objekten ab, entsprechen also den Verben. Es liegt deshalb nahe, die CARTRONIC®-Komponenten als UML-Klassen beziehungsweise UML-Objekte darzustellen.
- [0032] Über die Stereotypen «huelle», «informationsgeber», «koordinator» und «operator» werden die Komponentenklassen den oben eingeführten Kategorien zugeordnet. Die Komponente Umwelt aus Fig. 4 wird beispielsweise als Klasse mit dem Namen Umwelt und dem Stereotyp «informationsgeber» dargestellt.
- [0033] Die drei CARTRONIC®-Kommunikationsarten Auftrag, Abfrage und Anforderung fordern andere Komponenten auf, "etwas zu tun" – in objektorientierter Modellierung werden diese deshalb als Nachrichten interpretiert und mit den Stereotypen «auftrag», «abfrage» und «anforderung» als UML-Operationen modelliert. Aus dem CARTRONIC®-Auftrag moment\_ga wird die UML-Operation «auftrag» moment\_ga der Klasse Antrieb, aus der CARTRONIC®-Abfrage ?drehzahl an diese Komponente die UML-Operation «abfrage» drehzahl sowie aus der CARTRONIC®-Abfrage ?luftdruck an die Komponente Umwelt die UML-Operation «abfrage» luftdruck der Klasse Umwelt. Bei der Darstellung in Fig. 5 symbolisiert die doppelte Linie in den Klassen Antrieb, elektrisches Bordnetz und Umwelt, dass bislang noch keine Attribute vorhanden sind. Bei der Klasse Fahrzeug-Koordinator wird auf die Darstellung eventuell vorhandener Attribute oder Operationen ganz verzichtet und nur der Deklarationsbereich der Klasse abgebildet. Dies ergibt ein übersichtlicheres Diagramm. CARTRONIC®-Rückmeldungen können als Rückgabeparameter von UML-Operationen modelliert werden und sind deshalb bei der Abbildungsvorschrift nicht explizit als eigenständige UML-Operationen modelliert.
- [0034] Die Kapselung der einzelnen Komponenten mit dem Ziel einer definierten Zugriffsregelung und -Kontrolle wird über explizite UML-Schnittstellen für die UML-Operationen modelliert. Es wird zwischen Schnittstellen für Aufträge sowie Abfragen und Anforderungen unterschieden mit den beiden Stereotypen «interfaceAuftrag» und «interface». Die für Aufträge geltenden strenger Zugriffsregeln nach dem "Ein-Chef-Prinzip" werden hierdurch explizit modelliert. Fig. 6 zeigt dies am Beispiel der Klasse «operator» Antrieb.
- [0035] Eine tiefer detaillierte CARTRONIC®-Komponente zeigt Fig. 7 am Beispiel der Komponente Antrieb, hier im Unterschied zu Fig. 2 jedoch nur mit den drei Teilkomponenten Motor, Antriebs-Koordinator und Getriebe. UML-Aggregationen drücken eine "ist Teil von"-Beziehung aus und entsprechen somit exakt einer CARTRONIC®-Detaillierung. UML-Kompositionen drücken als Spezialfälle von UML-Aggregationen analog zum CARTRONIC®-Verständnis aus, dass Teilkomponenten ohne das Aggregat keinen Bestand haben. UML-Aggregat und UML-Komposition als logisches Ganzes delegieren (automatisch) Nachrichten, die sie erhalten, aber selber nicht ausführen können, an die entsprechende Teilkomponente (Analogie zur CARTRONIC®-Hülleneigenschaft). Fig. 8 zeigt die teilweise detaillierte UML-Klasse «huelle» Antrieb als UML-Komposition der UML-Klassen Motor, Antriebs-Koordinator und Getriebe.
- [0036] Große komplexe CARTRONIC®-Komponenten auf hohem abstraktem Niveau können analog zur Modellierung über UML-Klassen auch als UML-Subsysteme abgebildet werden. UML-Subsysteme sind eine Sonderform von UML-Paketen, die Schnittstellen besitzen dürfen und deshalb geeignet sind, eine sinnvolle Kapselung und Strukturabbildung im Großen zu gewährleisten. Die separaten Schnittstellen für Aufträge beziehungsweise Abfragen und Anforderungen bleiben dabei ebenso erhalten, wie analog die im Folgenden vorgestellten Vorgehensschritte. Verbindungen in Form einer Aggregation oder Komposition zwischen dem Subsystem und darin enthaltenen Modellelementen ermöglichen die Weiterleitung von Nachrichten von den Subsystem-Schnittstellen direkt zu den entsprechenden Komponenten. Wie in Fig. 9 gezeigt, kann eine Schnittstelle, hier «interfaceAuftrag» IA\_Antrieb, direkt durch die Schnittstelle eines internen Elements, hier «interfaceAuftrag» IA\_Antriebs-Koordinator, bereitgestellt werden.
- [0037] Die ständige Weiterentwicklung der CARTRONIC®-Modelle erfordert Veränderungs- und Austauschmöglichkeiten von Komponenten. Neben der im vorhergehenden Teilabschnitt beschriebenen Abbildung einzelner CARTRONIC®-Elemente in die UML müssen die Schritte und Vorgänge im fortlaufenden Entwicklungsprozess unterstützt werden. In der Praxis treten hierbei Mischformen der nachfolgend unterschiedenen Vorgehensschritte auf. Diese setzen sich zusammen aus der Detaillierung, der Abstraktion, dem Ersetzen sowie dem Löschen von Elementen.
- [0038] Bei der Detaillierung einer CARTRONIC®-Komponente entsteht nach dem CARTRONIC®-Regelwerk eine Hülle ohne eigene, logische Funktionen; alle bereits bekannten, existierenden Funktionalitäten werden auf die entstehenden Teilkomponenten verteilt. Die Kapselung beziehungsweise die Schnittstellen der zu detaillierenden Komponente müssen vollständig in der Hülle erhalten bleiben, um das Verhalten nach außen nicht zu verändern. Dies sind im Beispiel in Fig. 6 die Operationen moment\_ga und drehzahl der Klasse Antrieb. Fig. 10 zeigt die neue UML-Klasse nach der Detaillierung. Detailliert ist hier in die Klassen Motor, Antriebs-Koordinator und Getriebe sowie zusätzliche Operationen «auftrag» moment\_ma, «anforderung» rGang und «auftrag» einlegenGang. Zu beachten ist die Darstellung der Klasse Antrieb mit dem nunmehr leeren Operationsbereich sowie der Änderung des Stereotyps in «huelle». Die «abfrage» luftdruck der Komponente Motor aus der Komponente Antrieb heraus an den «informationsgeber» Umwelt wird modelliert über eine Beziehung zwischen diesen beiden Komponenten in Form einer Assoziation – beide müssen sich kennen – sowie eine Abhängigkeit von Motor an die entsprechende Schnittstelle von Umwelt. Dies gilt analog für alle CARTRONIC®-Kommunikationsbeziehungen. Die explizite Modellierung aller Beziehungen der UML-Klassen untereinander führt automatisch zu einem vollständigen Überblick über alle Zugriffe auf jede einzelne Schnittstelle.
- [0039] Abstraktionsvorgänge sind notwendig und sinnvoll für eine kompakte und übersichtliche Darstellung strukturierter, komplexer Komponentenstrukturen durch ihre Hüllenkomponente und Schnittstellen. Bei dem Vorgang der Abstraktion dürfen keine Informationen aus der detaillierten Darstellung verloren gehen, und Abhängigkeiten zwischen verschiedenen Komponenten müssen möglichst ohne Hinzufügung weiterer Information abgebildet werden. Bei der Ab-

straktion vom detaillierten UML-Aggregat Antrieb in Abb. 10 bleibt im Unterschied zu der in Fig. 6 dargestellten Komponente Antrieb eine "leere" Klasse (Fig. 11). Hierbei ergibt sich aber das Problem der Nicht-Sichtbarkeit der Beziehung zwischen einer Teilkomponente von Antrieb und Umwelt. Deshalb erfordert das UML-Modell zusätzlich zur CARTRONIC®-Funktionsarchitektur eine künstlich gesetzte Assoziation und Abhängigkeit zwischen den Komponenten Antrieb und Umwelt. Bei Einsatz eines Modellierungstools kann diese Problematik beispielsweise durch entsprechende Zusatzattribute berücksichtigt werden. 5

[0040] Die einfachste Art des Austauschs von Elementen ist ein Austausch mit gleicher Funktionalität, bei dem Modifikationen nur intern in einer Komponente ohne eine nach außen sichtbare Veränderung von Namen, Schnittstellen oder Beziehungen vorgenommen werden (Fig. 12).

[0041] Ein Austausch mit erweiterter Funktionalität liegt vor, wenn eine Komponente den Namen, vorhandene Beziehungen und ihre alte Funktionalität behält, jedoch um neue Funktionalität und damit neue Schnittstellenoperationen erweitert wird (Fig. 13). 10

[0042] Werden bei einem Austausch innere Elemente einer Komponente so modifiziert, dass neue Beziehungen zu anderen Komponenten erforderlich werden, handelt es sich um einen Austausch mit veränderten Beziehungen. Dies ist zum Beispiel der Fall bei zusätzlich abgefragter Information. Der Komponentennamen und die Schnittstellen bleiben hierbei unverändert (Fig. 14). 15

[0043] Beim Austauschen mit wegfallender Funktionalität entfallen einzelne Funktionen beziehungsweise Operationen in den Schnittstellen. Vor dem eigentlichen Löschvorgang muss deshalb eine (automatisierte) Konsistenzprüfung zur Sicherstellung erfolgen, dass keine weitere Komponente des Gesamtmodells die zu löschenden Elemente noch verwendet. 20

[0044] Das Löschen einer Komponente kann als Sonderform des Austauschs mit wegfallender Funktionalität betrachtet werden. Hierbei wird eine vorhandene Klasse in eine "leere" überführt durch iterativen Wegfall ihrer Operationen.

[0045] Es wird also ein Verfahren und Vorrichtung zur Modellierung eines mechatronischen Systems in einem Kraftfahrzeug im Rahmen eines objektbasierten Ordnungskonzepts als eine Abbildung in die Unified Modeling Language beschrieben. Diese stellt ein wesentliches Bindeglied zwischen der Analyse- und Design- beziehungsweise Entwurfsphase im objektorientierten Softwareentwicklungsprozess dar. Die Elemente der CARTRONIC® mit Komponenten und Hüllen als ihren Klassen beziehungsweise Objekten und Aufträgen (mit Rückmeldung), Abfragen (mit Hinweis) und Anforderungen als ihren Kommunikationsbeziehungen sind an Hand von Beispielen zusammen mit den wesentlichen Regeln aus dem Gesamtregelwerk vorgestellt. Für diese Modellierungselemente wird eine Abbildungsvorschrift in UML-Konstrukte dargestellt. CARTRONIC®-Komponenten einschließlich -Hüllen werden als UML-Klassen beziehungsweise -Objekte mit den Stereotypen «koordinator», «operator», «informationsgeber» und «huelle» abgebildet. Die CARTRONIC®-Kommunikationsbeziehungen Abfrage und Anforderung werden als UML-Operationen mit den Stereotypen «abfrage» und «anforderung» in Schnittstellen mit dem Stereotyp «interface» bereitgestellt, alle Auftragsoperationen werden durch den Stereotypen «auftrag» in Schnittstellen mit Stereotyp «interfaceAuftrag» gekennzeichnet. In eine detaillierte Komponente eingehende Nachrichten werden automatisch über Kompositionsbeziehungen an die jeweils zuständigen Teilkomponenten delegiert. Für nicht hierarchische Kommunikationsbeziehungen zwischen Auftraggeber und Beauftragtem wird zwischen diesen eine Assoziation und von dem Auftraggeber zu der Schnittstelle des Beauftragten eine Abhängigkeitsbeziehung modelliert. Die Kapselung von Komponenten wird somit über die beiden separaten Schnittstellen, die Komposition als implizite Zugehörigkeitsbeziehung und die Darstellung der Abhängigkeiten gewährleistet. Für die im Entwicklungsprozess auftretenden Vorgänge Detaillierung, Abstraktion, Austausch und Löschen von Komponenten werden Vorgehensweisen für den Einsatz eines kommerziell verfügbaren UML-Softwareentwicklungswerkzeuges aufgezeigt. 25 30 35 40

#### Patentansprüche

1. Verfahren zur Modellierung eines mechatronischen Systems in einem Kraftfahrzeug, wobei das mechatronische System aus mehreren Komponenten besteht, zwischen denen vorgegebene Kommunikationsbeziehungen bestehen, **dadurch gekennzeichnet**, daß die Komponenten und Kommunikationsbeziehungen zwischen den Komponenten im Rahmen einer objektorientierten Modellierungssprache dargestellt werden. 45 50

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die objektorientierte Modellierungssprache die unified modeling language ist.

3. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß auf der Basis der dargestellten Komponenten und Beziehungen das Softwaredesign erstellbar ist.

4. Verfahren zur Modellierung eines mechatronischen Systems in einem Kraftfahrzeug, wobei das mechatronische System in mehrere Komponenten zwischen denen vorgegebene Kommunikationsbeziehungen wie Abfrage, Anforderung und Auftrag bestehen, gegliedert ist, dadurch gekennzeichnet, daß eine Abbildungsvorschrift vorgesehen ist, mit der die Komponenten und Beziehungen des Systems in Konstrukte einer objektorientierten Modellierungssprache, vorzugsweise der UML-Sprache, umgesetzt werden. 55

5. Verfahren nach Anspruch 4, dadurch gekennzeichnet, dass die Abbildungsvorschrift darin besteht, dass die Komponenten einschließlich -Hüllen als UML-Klassen beziehungsweise -Objekte mit den Stereotypen «koordinator», «operator», «informationsgeber» und «huelle» abgebildet werden, die Kommunikationsbeziehungen Abfrage und Anforderung als UML-Operationen mit den Stereotypen «abfrage» und «anforderung» in Schnittstellen mit dem Stereotyp «interface» bereitgestellt werden und alle Auftragsoperationen durch den Stereotypen «auftrag» in Schnittstellen mit Stereotyp «interfaceAuftrag» gekennzeichnet werden. 60 65

6. Vorrichtung zur Modellierung eines mechatronischen Systems in einem Kraftfahrzeug, welches aus mehreren Komponenten besteht, zwischen denen vorgegebene Kommunikationsbeziehungen bestehen, dadurch gekennzeichnet, dass die Komponenten und Kommunikationsbeziehungen zwischen den Komponenten im Rahmen einer

# DE 100 15 114 A 1

objektorientierten Softwaresprache dargestellt sind.

7. Vorrichtung zur Modellierung eines mechatronischen Systems in einem Kraftfahrzeug, welches aus mehreren Komponenten besteht, zwischen denen vorgegebene Kommunikationsbeziehungen bestehen, gekennzeichnet durch eine Abbildungsvorschrift, mit der die Komponenten und Beziehungen des Systems in Konstrukte einer objektorientierten Modellierungssprache, vorzugsweise der UML-Sprache, umgesetzt werden.

---

Hierzu 8 Seite(n) Zeichnungen

---

10

15

20

25

30

35

40

45

50

55

60

65



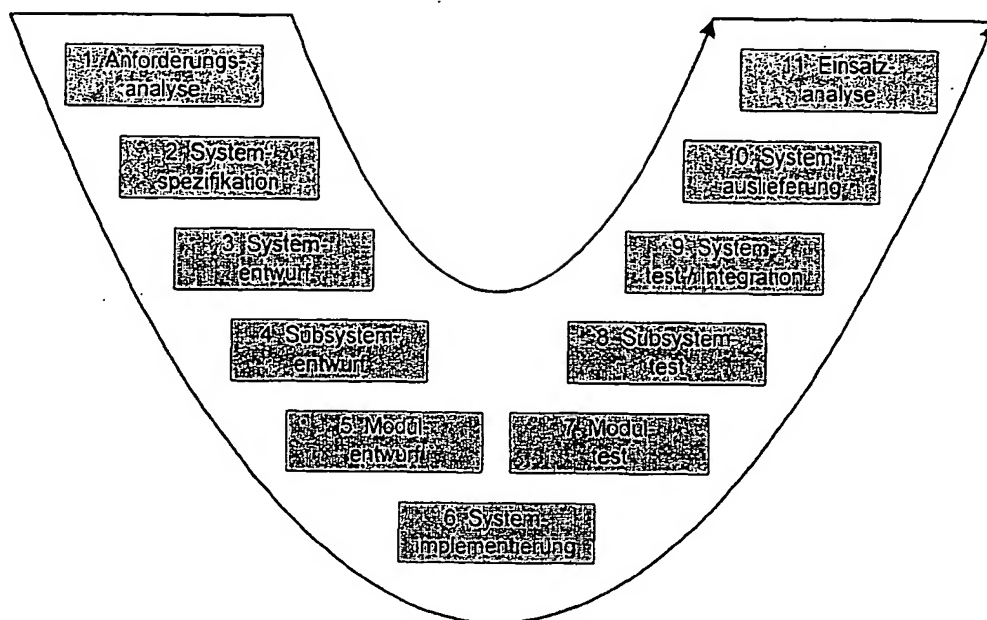


Fig. 1

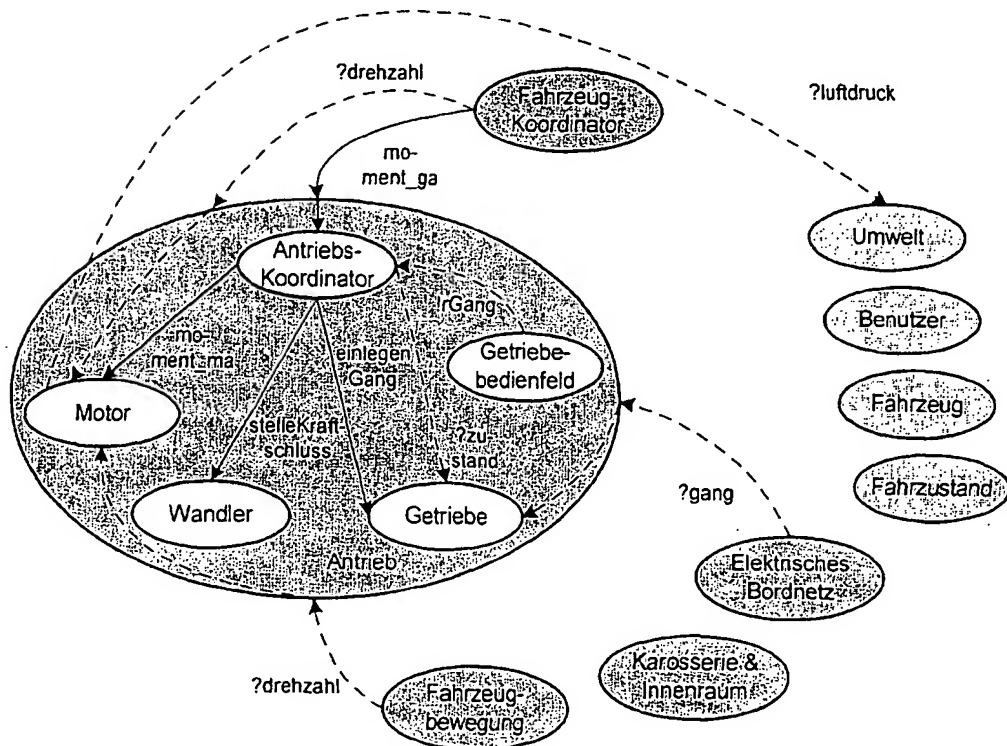


Fig. 2

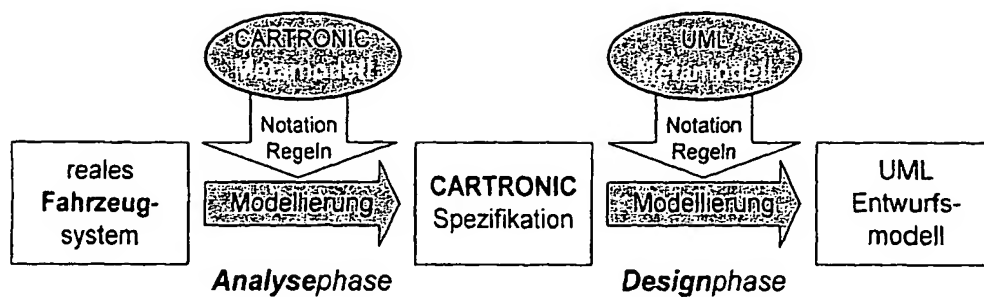


Fig. 3

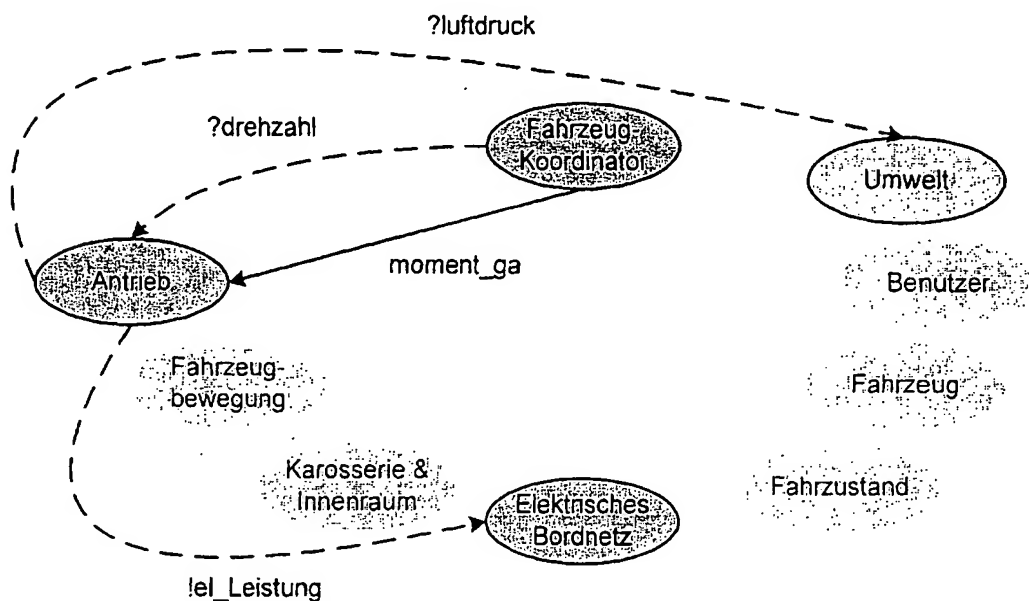


Fig. 4

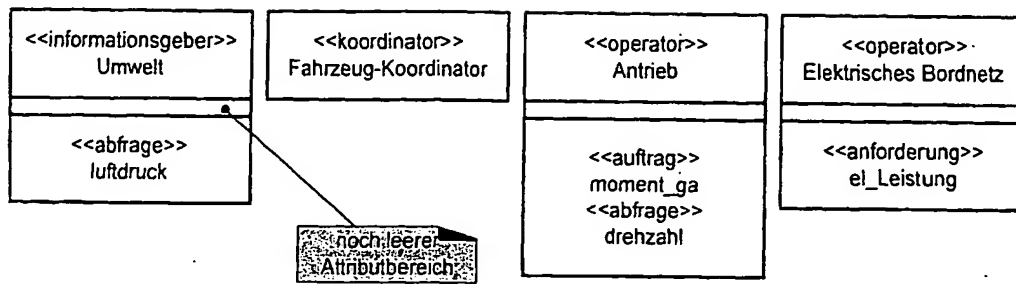


Fig. 5

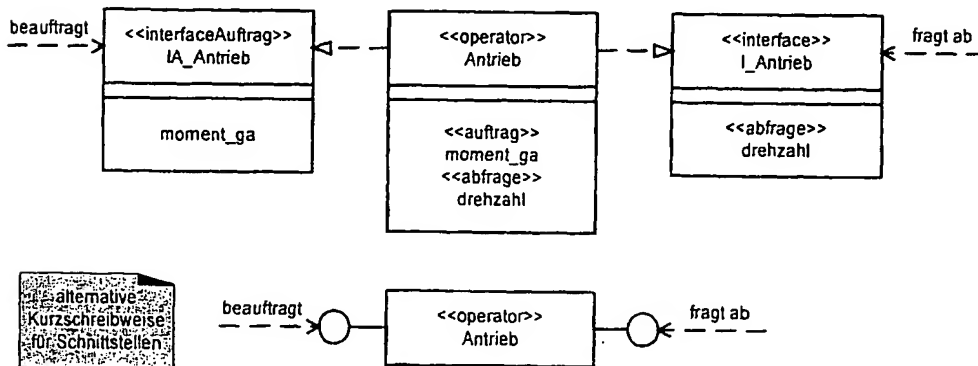


Fig. 6

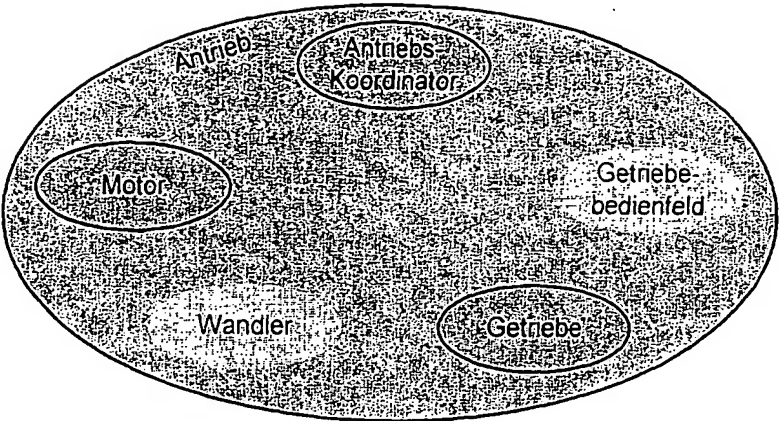


Fig. 7

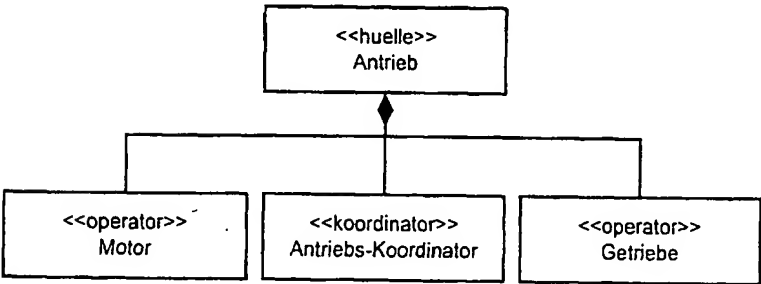


Fig. 8

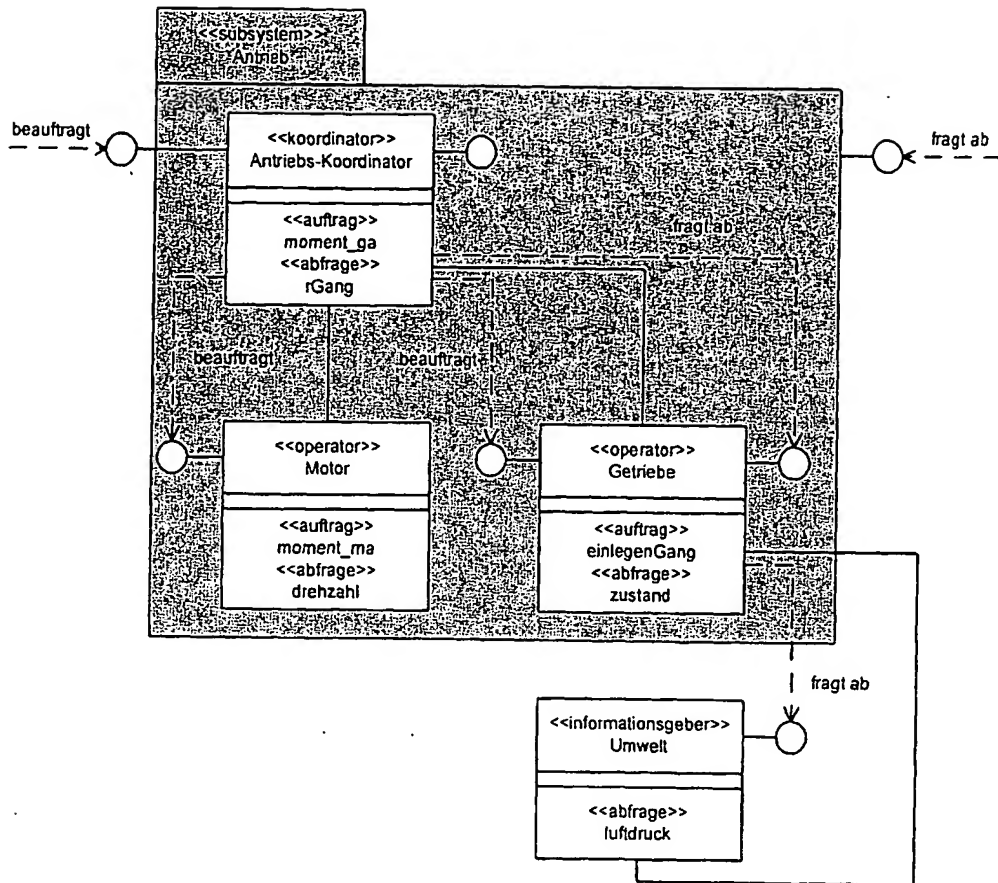


Fig. 9

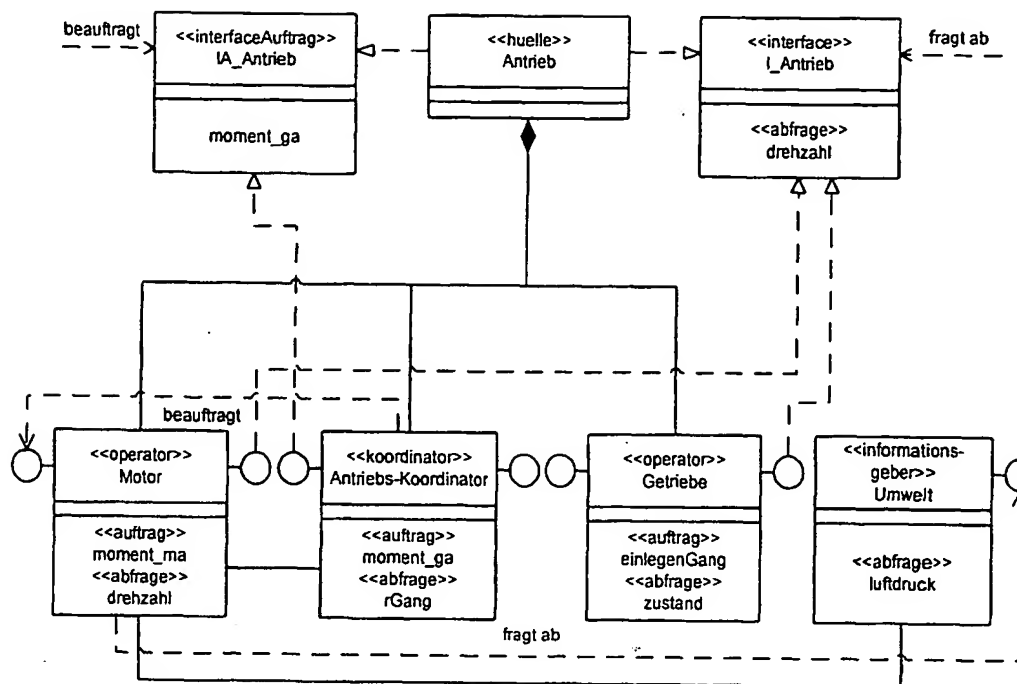


Fig. 10

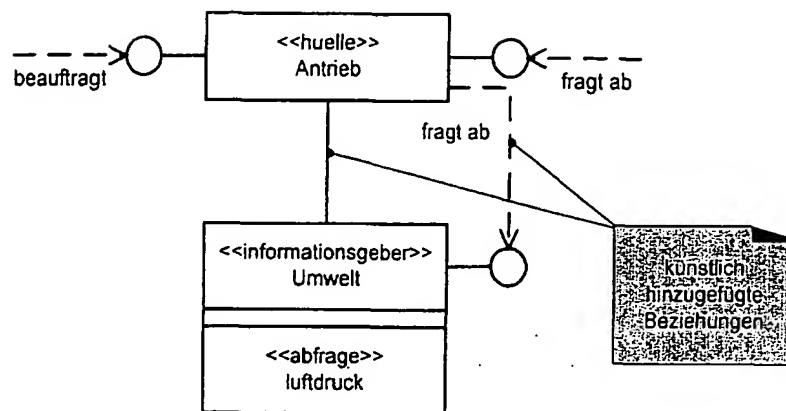


Fig. 11

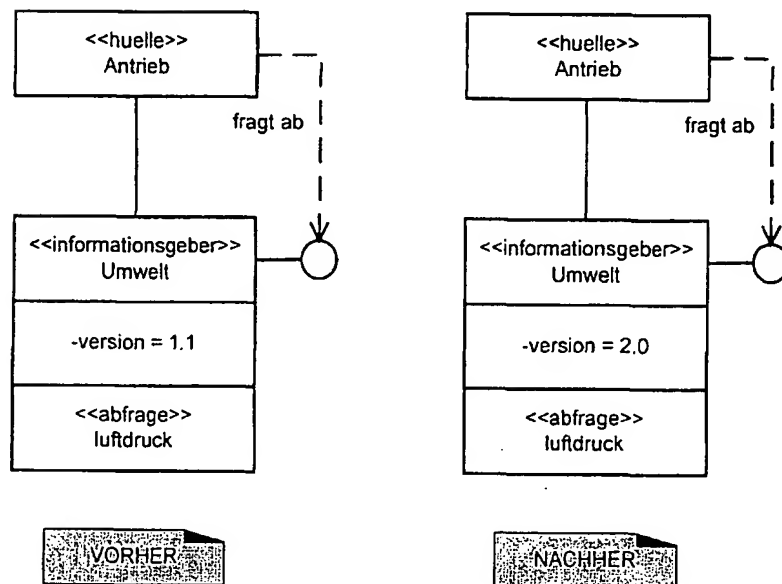


Fig. 12

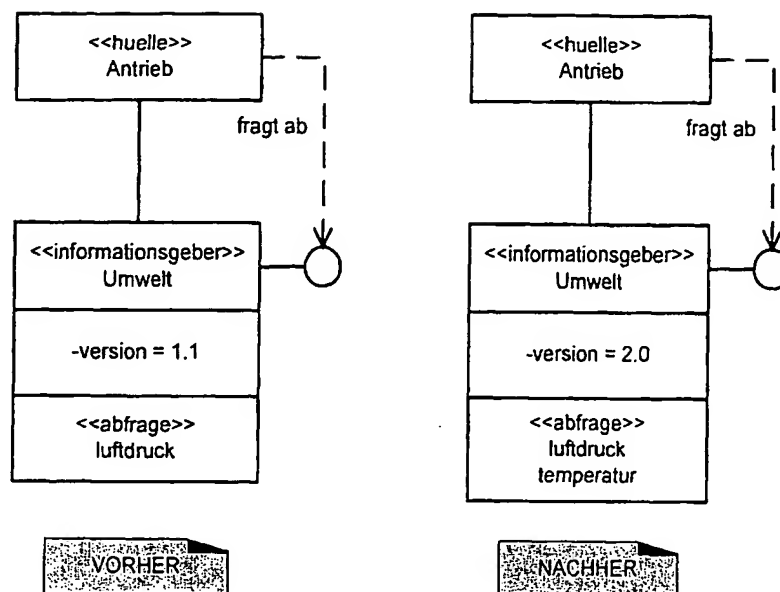


Fig. 13

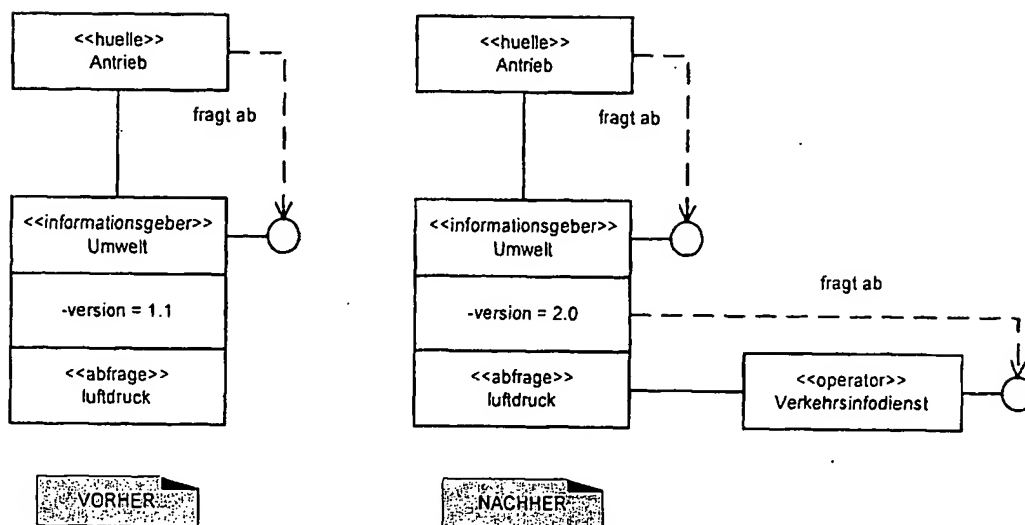


Fig. 14



# METHOD AND DEVICE TO DESIGN A DOMAIN MODEL FOR CONTROL SYSTEMS IN VEHICLES WITH RESPECT OF THE FUNCTIONAL REQUIREMENTS

Publication number: DE10015114

Publication date: 2001-10-04

Inventor: FLORES PIO TORRE (DE); SCHIRMER JUERGEN (DE); WALTHER MICHAEL (DE); HUELSEH HOLGER (DE); BERTRAM TORSTEN (DE); HECKES MARC (DE); PETERSEN JOERG (DE)

Applicant: BOSCH GMBH ROBERT (DE)

Classification:

- international: B60R16/02; B60R16/023; B62D65/00; G06F9/44; B60R16/02; B60R16/023; B62D65/00; G06F9/44; (IPC1-7): G06F17/50; B62D65/00

- European: B62D65/00; B60R16/023D

Application number: DE20001015114 20000328

Priority number(s): DE20001015114 20000328

Also published as:

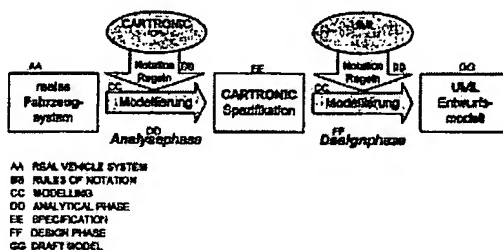
WO0173279 (A3)  
WO0173279 (A2)  
WO0172552 (A3)  
WO0172552 (A2)  
US7188016 (B2)

more >>





Report a data error here

## Abstract of DE10015114

The invention relates to a method and a device for modelling a mechatronic system in a motor vehicle according to an object-based architecture, whereby said system is represented in the unified modelling language. The elements of the CARTRONIC(R) which comprise components and sleeves as the classes thereof or objects and orders (with feedback), inquiries (with hints) and requests as the communications relations thereof are provided by means of examples and together with the essential rules of the entire catalogue of rules. A representation rule for said modelling elements is provided in UML constructs.



Data supplied from the esp@cenet database - Worldwide

**Ind. plant atomisation design system using combined process and control model****Publication number:** DE19639424**Publication date:** 1997-03-27**Inventor:** SONST HORST (DE); FISCHER HORST (DE);  
LOEWEN ULRICH DR (DE); FREITAG HARTMUT DR  
(DE); WEHN NORBERT DR (DE); ROSEN ROLAND  
(DE)**Applicant:** SIEMENS AG (DE)**Classification:****- international:** G05B17/02; G05B19/042; G06F17/50; G05B17/00;  
G05B19/04; G06F17/50; (IPC1-7): G06F17/50**- European:** G05B17/02; G05B19/042P; G06F17/50C**Application number:** DE19961039424 19960925**Priority number(s):** DE19961039424 19960925; DE19951035549 19950925**Also published as:** WO9712301 (A1)  
 EP0852759 (A1)  
 EP0852759 (A0)  
 EP0852759 (B1)[Report a data error here](#)**Abstract of DE19639424**

The design system uses a combined model defining both the ind. process characteristics of the plant and the control engineering characteristics. The model is used for control and/or regulation and/or simulation of the plant operation for failure analysis. Pref. the technological specifications are selected and a corresponding technical design solution is provided, using a file (11) of component types, with a simulation process used for validation.

---

Data supplied from the esp@cenet database - Worldwide



⑮ **BUNDESREPUBLIK  
DEUTSCHLAND**



**DEUTSCHES  
PATENTAMT**

⑫ **Offenlegungsschrift**  
⑩ **DE 196 39 424 A 1**

⑤① Int. Cl.<sup>8</sup>:  
**G 06 F 17/50**

⑳ Aktenzeichen: 198 39 424.4  
㉔ Anmeldetag: 25. 9. 96  
㉕ Offenlegungstag: 27. 3. 97

**DE 196 39 424 A 1**

⑤④ Innere Priorität: ㉔ ㉕ ㉖

25.09.95 DE 195355490

⑤⑦ Anmelder:

Siemens AG, 80333 München, DE

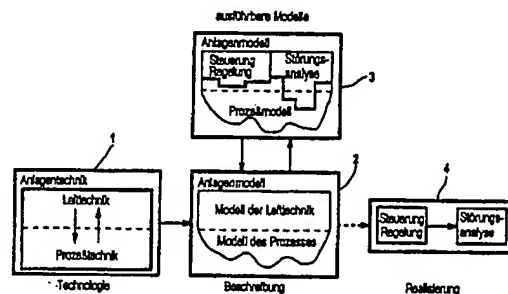
⑤⑦ Erfinder:

Sonst, Horst, 91093 Heßdorf, DE; Fischer, Horst,  
90475 Nürnberg, DE; Löwen, Ulrich, Dr., 91052  
Erlangen, DE; Freitag, Hartmut, Dr., 91230 Happurg,  
DE; Wehn, Norbert, Dr., 85579 Neubiberg, DE;  
Rosen, Roland, 82211 Herrsching, DE

⑤④ Entwurfsverfahren für die Anlagentechnik und rechnergestütztes Projektierungssystem zur Verwendung bei diesem Verfahren

⑤⑦ Entwurfsverfahren für die Anlagentechnik und rechnergestütztes Projektierungssystem zur Verwendung bei diesem Verfahren.

Insbesondere für die Automatisierung von Industrieanlagen läßt sich die Technologie der Anlage durch Prozeßtechnik einerseits sowie durch Leittechnik andererseits beschreiben. Gemäß der Erfindung werden aus Modellen des Prozesses und der Leittechnik verfahrenstechnische Objekte gebildet, die in einem rechnergestützten Projektierungssystem verarbeitet werden, woraus ineinander verzahnte Modelle mit Steuerung und/oder Regelung und/oder Simulation einschließlich Störungsanalyse erzeugt werden. Beim zugehörigen Projektierungssystem erfolgt mit einem Rechner (10) gleichermaßen ein Zugriff auf eine Komponentenbibliothek (11) von verfahrenstechnischen Elementen und auf eine Beschreibung (12) der Elemente der konkreten Anlage aus der Sicht eines verfahrenstechnischen Technologen.



**DE 196 39 424 A 1**

Entwurfsverfahren für die Anlagentechnik und rechnergestütztes Projektierungssystem zur Verwendung bei diesem Verfahren

Die Erfindung bezieht sich auf ein Entwurfsverfahren für die Anlagentechnik, insbesondere zur Verwendung bei der Automatisierung von Industrieanlagen, wobei die Technologie der Anlage durch Prozeßtechnik einerseits und durch Leittechnik andererseits beschrieben wird. Daneben bezieht sich die Erfindung auch auf ein rechnergestütztes Projektierungssystem zur Verwendung bei diesem Verfahren, enthaltend einen Digitalrechner mit Zentraleinheit, Arbeits-, Programm- und Datenspeichern und mit Mitteln zur Codegenerierung.

Bei der Automatisierung von Industrieanlagen wird bisher im allgemeinen von einer vorgegebenen Prozeßtechnik ausgegangen, zu welcher eine zugehörige Leittechnik entwickelt werden muß, was üblicherweise von einem Elektrotechniker als Projektteur durchgeführt wird.

Eine wesentliche Aufgabe bei der Konzeption der Leittechnik einer Anlage ist zunächst einmal eine Aufgabenklärung, die zusammen mit dem späteren Betreiber der Anlage erfolgen muß. Wesentlich ist dabei die Erstellung einer externen Spezifikation, welche die Anforderungen an die zu projektierende Leittechnik beschreibt. Üblich ist heutzutage, daß dazu ein Technologe mit dem Projektteur zusammenarbeitet. Als Ergebnis entstehen dabei Funktionspläne, d. h. eine Beschreibung der zu lösenden Projektierungsaufgabe aus der Sicht des Elektrotechnikers.

Bei der Umsetzung solcher Funktionspläne in eine weiterbearbeitbare softwaremäßige Lösung gibt es jedoch häufig Probleme, weil Diskrepanzen zwischen externer Spezifikation und der fertigen Lösung entstehen.

Aufgabe der Erfindung ist es daher, die bisher bekannten Entwurfsverfahren für die Anlagentechnik zu verbessern und zugehörige Werkzeuge zu schaffen.

Die Aufgabe wird erfindungsgemäß durch folgende Maßnahmen gelöst:

- Es werden aus Modellen der Prozesse und der Leittechnik verfahrenstechnische Objekte gebildet,
- die verfahrenstechnischen Objekte werden in einem rechnergestützten Projektierungssystem verarbeitet,
- daraus werden ineinander verzahnte Modelle bestehend aus Steuerung und/oder Regelung und/oder Simulation einschließlich Störungsanalyse erzeugt.

Bei der Erfindung ist das Verfahrensergebnis als Software und/oder als Hardware ausführbar. Wesentlich ist, daß zur Beschreibung der Anlage jeweils rechnergestützt die Modelle des Prozesses und der Leittechnik in einem Anlagenmodell zusammengefaßt und daraus automatisiert ineinander verzahnte Modelle erstellt werden.

Im Rahmen der Erfindung werden jeweils anwendungsneutrale Beschreibungsmittel gewählt. Insbesondere kommt es darauf an, die Spezifikation so technologienah zu wählen, daß die verfahrenstechnische Lösung erkennbar ist, wobei die Funktionsbeschreibung der Strukturbeschreibung untergeordnet ist. Solche Spezifikationen können dann formal analysiert und simulativ validiert werden. Damit ist es möglich, unmittelbar aus der Spezifikation einen Code für das im Rahmen des

Projektierungssystem zu verwendende Automatisierungsgerät zu erzeugen.

Besonders vorteilhaft ist beim erfindungsgemäßen Verfahren, daß die bei der Projektierung entwickelten Modelle anschließend unmittelbar bei der Anlagenrealisierung verwendet werden. Im Ergebnis ergibt sich dadurch ein wesentliches Einsparpotential.

Bei einem geeigneten Projektierungssystem zur Verwendung bei dem erfindungsgemäßen Entwurfsverfahren erfolgt mit dem Digitalrechner gleichermaßen ein Zugriff auf eine Bibliothek von Komponententypen und auf eine Beschreibung der Komponenten der konkreten Anlage. Entscheidend ist dabei, daß für die Benutzung der Komponentenbibliothek die Beschreibung aus der Sicht des verfahrenstechnischen Technologen definiert ist. Vorzugsweise hat die Komponentenbibliothek eine branchenspezifische Auslegung.

Mit dem erfindungsgemäßen Projektierungssystem ist also ein Werkzeug geschaffen, bei dem die Komponententypen aus der Sicht des Technologen definiert sind und ein unmittelbarer Bezug zur Prozeßtechnik besteht.

Die Erfindung wurde beispielhaft im Rahmen der Papiertechnologie untersucht und es wurde ein Demonstrator für eine Altpapieraufbereitung erstellt. Darüber hinaus konnte gezeigt werden, daß die erzielten Ergebnisse auf andere Bereiche der Anlagentechnik verallgemeinerbar sind.

Weitere Einzelheiten und Vorteile der Erfindung ergeben sich aus der nachfolgenden Figurenbeschreibung von Ausführungsbeispielen anhand der Zeichnung. Es zeigen jeweils als Prinzipdarstellung

Fig. 1 das neue Entwurfsverfahren,

Fig. 2 das zugehörige Projektierungssystem,

Fig. 3 der hierarchische Aufbau der dabei als Komponententypen verwendeten verfahrenstechnischen Elemente,

Fig. 4 die Beschreibung einer Anlage mit dem Projektierungssystem,

Fig. 5 die dabei verwendete Methode zur automatischen Codegenerierung und

Fig. 6 das Vorgehen bei der Fehlerdiagnose.

Unter einer Anlage versteht man im allgemeinen die Gesamtheit der Ausrüstungen eines Betriebs, die zur Produktion, zur Fertigung, zur Energieerzeugung und/oder zu Förder- bzw. Transportzwecken erforderlich sind. Im Nachfolgenden werden unter dem Begriff "Industrieanlage" überwiegend Großanlagen wie beispielsweise Kraftwerke, Müllverbrennungsanlagen oder auch Papierfabriken verstanden.

Zur Erstellung einer Anlage müssen unterschiedliche Partner zusammenwirken: Dies ist einmal der Betreiber, dessen Ziel es ist, durch das Anbieten von Dienstleistungen bzw. Produkten ein Ergebnis zu erzielen. Dazu orientiert er sich an den Bedürfnissen des Marktes. Er formuliert seine Ziele mit Hilfe von Begriffen wie Durchsatz, Kosten, Qualität und/oder Produktivität.

Vom Betreiber der künftigen Anlage nehmen üblicherweise Technologen die Anforderungen entgegen und setzen sie um in einen Entwurf für eine Anlagenkonfiguration. Technologen benötigen dabei Wissen über notwendige Verfahrensschritte, da sie für die Auslegung der verschiedenen Anlagenteile zuständig sind. Als Ergebnis erzeugen die Technologen ein Technologieschema, in dem die einzelnen Maschinen, Sensoren, Stellglieder, aber auch Material-, Wirk- und Informationsflüsse festgelegt sind.

Da für gewisse Verfahrensschritte der Anlage Einzel-

lösungen in Form von Maschinen und Ausrüstungen, die von Komponententierlieferanten angeboten werden, bereits vorhanden sind, ist es zweckmäßig, zwischen Lieferanten für die Prozeßtechnik und Lieferanten für die Leittechnik zu unterscheiden. Üblicherweise erstellen Systemintegratoren aus den einzelnen Komponenten der Komponententierlieferanten die komplette Anlage.

Wesentlich für ein erfolgreiches Entwurfsverfahren sowie die weitere Erstellung bei der Anlagentechnik ist die technologische Kompetenz, welche für die Leittechnik einerseits und für die Prozeßtechnik andererseits maßgebend ist. Die Leittechnik umfaßt dabei alle Einrichtungen, die zum automatischen Betrieb der Anlage notwendig sind. Dazu gehören beispielsweise unter anderem die Aktoren und Sensoren, die Hard- und Software der Automatisierungsebene und die Komponenten der Leit- und Führungstechnik der Anlage. Die eigentliche Prozeßtechnik wird dagegen durch Aggregate, Pumpen, Ventile u. dgl. beschrieben.

In Fig. 1 geht der Block 1 von der bekannten Technologie einer zu projektierenden Anlage aus. Für die Anlagentechnik ist dabei sowohl die Leittechnik als auch die Prozeßtechnik von Bedeutung, welche miteinander in Wechselwirkung stehen. Anhand von Block 2 wird die zugehörige Beschreibung verdeutlicht: Das Anlagenmodell besteht dabei einerseits aus einem Modell der Leittechnik und andererseits aus einem Modell des Prozesses, welche zumindest einmal voneinander zu trennen sind. Im Block 3 sind die ausführbaren Modelle eingetragen, die im einzelnen miteinander verzahnt sind. Dabei verdeutlicht die Verzahnung, daß nunmehr erstmalig die bisher üblichen Abgrenzungen aufgehoben sind. Insbesondere die Teilmodelle greifen jeweils auf solche Informationen, die in anderen Teilmodellen bereits beschrieben sind, zurück. Eine doppelte Beschreibung unterbleibt also, was insbesondere bei einer im allgemeinen immer notwendigen Störungsanalyse Vorteile hat.

Gemäß Fig. 1 sind dem Prozeßmodell entsprechende Einheiten zur Steuerung und/oder Regelung und/oder Simulation einerseits sowie zur Störungsanalyse andererseits zugeordnet. Schließlich umfaßt der Block 4 die spätere Realisierung der Anlage mit entsprechender Störungsanalyse.

Bei der Vorgehensweise entsprechend Fig. 1 ist die Basis des Entwurfsverfahrens eine technologiennahe Beschreibung der Struktur einer Anlage und deren Funktionsweise. Durch die Kombination von Techniken zum Entwurf von Steuerungen und Regelungen, zur Simulation und zur Diagnose wird ein ausführbares Anlagenmodell erstellt, das den Spezifikationen entspricht.

Die Spezifikation für das Anlagenmodell kann anschließend im einzelnen formal analysiert und simulativ validiert werden, d. h. durch Simulationsrechnungen bestätigt werden. Die Simulation erfolgt mit graphischer Unterstützung, um gemeinsam dem Technologen die Aufgabenklärung zu verdeutlichen und ggfs. die Spezifikation zu modifizieren. Die validierte Spezifikation wird dahingehend genutzt, daß daraus weitestgehend automatisch ein Code für Automatisierungssysteme erzeugt werden kann, was im einzelnen in der europäischen Patentveröffentlichung EP-A-0 671 027 für spezifische Anwendungszwecke bei der Bahntechnik beschrieben ist.

Ein System zur Verwendung bei einer Vorgehensweise gemäß Fig. 1 beinhaltet eine technologiennahe Beschreibung der betreffenden Anlage. Dabei setzt sich die konkrete Anlage zusammen aus Objekten der Komponententierbibliothek und wird so auf die abzubildende

Anlage über Parameter und Wechselbeziehungen eingestellt. Wichtig ist, daß die Komponententypen aus der Sicht des Technologen definiert wurden, so daß nunmehr ein unmittelbarer Bezug zur Prozeßtechnik möglich ist. Für die einzelnen Komponententypen wird die Struktur gemeinsam für alle Anteile der Entwurfsverfahren und das Verhalten unter den Aspekten Steuerung und Regelung, Prozeßsimulation und Störungsanalyse spezifiziert. Die Komponenten der konkreten Anlage erhält man daraus durch Instanziierung der Komponententypen.

In Fig. 2 kennzeichnet 10 einen üblichen Digitalrechner mit Zentraleinheit 101, Eingabeeinheit 102 und Monitor 103. Die Zentraleinheit 101 beinhaltet u. a. Arbeits-, Programm- und Datenspeicher, die nicht im einzelnen dargestellt sind. Wesentlich ist im vorliegenden Zusammenhang, daß im Datenspeicher eine Komponententypbibliothek 11 von verfahrenstechnischen Elementen vorhanden ist, die sich zu komplexeren Objekten kombinieren lassen. Letzteres ist in Fig. 2 durch den separaten Block 11, der mit der Zentraleinheit 101 bzw. dem Monitor 103 verbunden ist, angedeutet. Durch Instanziierung einzelner Elemente lassen sich im Bereich 11' solche Objekte als Modelle bilden und unmittelbar auf dem Monitor 103 darstellen.

In Fig. 3 ist verdeutlicht, daß die Komponententierbibliothek 12 einen hierarchischen Aufbau hat. Beispielsweise sind in einer untersten Ebene 111 Sensoren und Aktoren vorhanden und in der nächsten Ebene 112 Ventile und Motoren als sog. Typicals. In den darüberliegenden Ebenen 113 bis 116 folgen komplexere Komponenten sowie Teilanlagen, Anlagen und die Fabrik als komplette Industrieanlage. Solche Objekte können vom Benutzer unmittelbar abgerufen werden.

In Fig. 4 stellt Block 11 eine solche Komponententierbibliothek dar, die branchenspezifisch strukturiert ist, und Block 12 eine zugehörige zu projektierende Industrieanlage. Die Beschreibung der konkreten Anlage 12 entsprechend der Vorgehensweise gemäß Fig. 1 erfolgt durch Instanziierung der Komponenten aus der Komponententierbibliothek 11. Durch Hinzufügen der Parameterwerte mit entsprechenden Wechselbeziehungen der Anlage 12 läßt sich die Beschreibung selbst realisieren. Durch die Berücksichtigung der Anlagenstruktur wird dabei für die einzelnen Anteile des Entwurfsverfahrens das globale Verhalten der Anlage aus den lokalen Beschreibungen automatisch konstruiert.

Die Codegenerierung für die Steuerung und Regelung einer zu projektierenden Anlage kann entweder unmittelbar auf der Beschreibung der Anlage aufsetzen oder alternativ auf dem ausführbaren Modell der Steuerung und Regelung basieren. Gemäß letzterer Alternative ist es dafür erforderlich, daß die Steuerungsvorgänge explizit deterministisch beschrieben sind. Es wird in diesem Fall von einer alternativen Codegenerierung gesprochen.

Die alternative Codegenerierung nutzt gewisse Phasen eines bekannten CSL-Compilers, in dem sie nicht auf der bekannten Sprache CSL, sondern auf der Sprache SCSL aufsetzt. Es wird dabei nicht der volle Umfang der Sprache SCSL von der alternativen Codegenerierung berücksichtigt. Hierzu ist in Fig. 5 ein sogenannter CSL-Compiler 30 dargestellt, dem aus Fig. 1 bis Fig. 4 die Blöcke der Komponententierbibliothek 11 und des konkreten Anlagenaufbaus 12 zugeordnet sind. Der CSL-Compiler 30 besteht aus den Einheiten 31 für die Instanziierung, 32 für die sogenannte Expansion, 33 für die Codierung und 34 für das Gleichungslösen, mit denen

ein Automat 40 generiert wird. Bei der alternativen Cod degenerierung wird an der Einheit 32 abgegriffen und über die SCSL-Sprache in der Einheit 36 zur Cod degenerierung der tatsächliche Code realisiert.

Bei der Sprache SCSL wird also vom CSL-Compiler 30 ein Zwischenergebnis ausgenutzt, das nach der Expansion erzeugt wird. Als Zielsprache wird bei der Cod degenerierung ein zielmaschinenunabhängiger Zwischencode erzeugt, der der Norm IEEC 1131-3 entspricht.

Das an den Prinzipschaubildern gemäß Fig. 1 und 3 erläuterte Verfahren geht aus von Verfahren zur Beschreibung und Analyse zustandsähnlicher Systeme. Dieses Verfahren ermöglicht die Konstruktion der Steuerung und Regelung einer hierarchisch aufgebauten Anlage aus der Strukturbeschreibung und der Beschreibung lokaler Steuerungs- und Regelungsvorgänge. Damit werden die Inkonsistenzen zwischen den unterschiedlichen lokalen Anforderungen automatisch aufgedeckt. Die Regelungsaufgaben sind dagegen weitgehend entkoppelt, so daß eine Trennung in analoge und diskrete Zusammenhänge möglich ist.

Für eine simulative Validation der Steuerung- und Regelungsvorgänge wird das Verhalten von Prozeß, Sensorik und Aktorik in Form eines Simulationsmodells beschrieben. Das Gesamtverhalten der Anlage wird auch hier durch eine Kombination lokaler Modelle unter Berücksichtigung der Strukturbeschreibung konstruiert. Für die Validation des Diagnoseanteils werden zusätzlich mögliche Fehlerfälle von Komponenten modelliert.

Zur Analyse von Störungen, die durch Anlagenfehler hervorgerufen werden, lassen sich Verfahren der modellbasierten Diagnose verwenden. Dafür wird die Kombination lokaler Modelle des Normal- und Fehlverhaltens von Komponenten entsprechend der Anlagenstruktur zur Erkennung und Identifikation von Fehlern benutzt. Für eine engere Integration von Steuerungsentwurf und Diagnosetechnik sind Konzepte zur Fehlererkennung durch die Steuerung und Analyse des Steuerungszustandes bekannt.

Aus Fig. 6 ergibt sich im wesentlichen selbsterklärender Weise, wie die Diagnose durch Finden und Analyse von Diskrepanzen erfolgen kann. Dabei wird davon ausgegangen, daß die Erkennung des Zielverhaltens eines technischen Systems die Bestimmung plausibler Erläuterungen für das beobachtete Fehlverhalten, d. h. die Diagnose, für die Wiederherstellung seiner ursprünglichen Funktion entscheidend ist. Die Diagnose setzt dabei das Wissen über das normale erwartete Verhalten des Systems voraus.

Bei der modellbasierten Diagnose gemäß Fig. 4 wird von einem expliziten Modell des zu diagnostizierenden Systems ausgegangen. Dies Modell umfaßt eine Beschreibung der Struktur des Systems, d. h. der Komponenten und ihrer Verbindungen, sowie lokale, d. h. kontextfreie Beschreibungen des Verhaltens einzelner Komponenten. Es werden im wesentlichen qualitative Modelle verwendet, um den Modellierungsprozeß zu vereinfachen.

Auf der Basis einer solchen Systembeschreibung wird unter der Annahme, daß jede Komponente sich korrekt verhält, das Verhalten des Gesamtsystems vorhergesagt und dabei protokolliert, von welchen Korrektheitsannahmen bestimmte Aspekte dieses Verhaltens abhängen. Treten Diskrepanzen zwischen vorhergesagtem und beobachtetem Verhalten auf, werden die protokollierten Abhängigkeiten verwendet, um die Mengen von

Korrekturannahmen zu identifizieren, die für die Diskrepanzen verantwortlich ist. Diese Mengen von inkonsistenten Annahmen werden auch Konflikte genannt. Um alle Konflikte aufzulösen, muß eine Diagnose mindestens eine Annahme aus jedem Konflikt zurückziehen. Auf diese Weise erhält man im allgemeinen mehrere Diagnosen, die sich durch genauere Modelle und zusätzliche Beobachtungen weiter modifizieren lassen.

#### Patentansprüche

1. Entwurfsverfahren für die Anlagentechnik, insbesondere zur Verwendung bei der Automatisierung von Industrieanlagen, wobei die Technologie der Anlage durch Prozeßtechnik einerseits und durch Leittechnik andererseits beschrieben wird, gekennzeichnet durch folgende Maßnahmen:

- Es werden aus Modellen der Prozesse und der Leittechnik verfahrenstechnische Objekte gebildet,
- die verfahrenstechnischen Objekte werden in einem rechnergestütztem Projektierungssystem verarbeitet,
- daraus werden ineinander verzahnte Modelle bestehend aus Steuerung und/oder Regelung und/oder Simulation einschließlich Störungsanalyse erzeugt.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß das Verfahrensergebnis als Software und/oder als Hardware realisierbar ist.

3. Verfahren nach Anspruch 1, wobei eine externe Spezifikation vorgegeben ist, dadurch gekennzeichnet, daß die Spezifikation technologienah gewählt wird, so daß sich solche verfahrenstechnische Objekte ergeben, bei denen die Funktionsbeschreibung der Strukturbeschreibung untergeordnet ist.

4. Verfahren nach Anspruch 3, dadurch gekennzeichnet, daß die Spezifikation formal analysiert und simulativ validiert wird.

5. Verfahren nach Anspruch 4, dadurch gekennzeichnet, daß aus der Spezifikation ein zielmaschinenneutraler Code für Automatisierungssysteme erzeugt wird.

6. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß zur Benutzungsanalyse eine modellbasierte Diagnose durchgeführt wird.

7. Verfahren nach Anspruch 6, dadurch gekennzeichnet, daß bei der modellbasierten Diagnose eine Kombination lokaler Modelle angewandt wird.

8. Verfahren nach Anspruch 6 und 7, dadurch gekennzeichnet, daß bei Störungen, die durch Anlagenfehler hervorgerufen werden, Modelle des Normal- und Fehlverhaltens von Komponenten entsprechend der Anlagenstruktur herangezogen werden.

9. Verfahren nach Anspruch 8, dadurch gekennzeichnet, daß die Diagnosetechniken in den Anlagenentwurf integriert sind.

10. Rechnergestütztes Projektierungssystem zur Durchführung des Verfahrens nach Anspruch 1 oder einem der Ansprüche 2 bis 9, enthaltend einen Digitalrechner mit Zentraleinheit, Arbeits-, Programm- und Datenspeichern und mit Mitteln zur Cod degenerierung, dadurch gekennzeichnet, daß mit dem Digitalrechner (10) gleichermaßen ein Zugriff auf eine Bibliothek (11) von verfahrenstechnischen Komponententypen und auf eine Beschreibung (13) der Elemente der konkreten Anlage (12)

erfolgt.

11. Projektierungssystem nach Anspruch 10, dadurch gekennzeichnet, daß die Komponentenbibliothek (11) Bestandteil des Datenspeichers ist.

12. Projektierungssystem nach Anspruch 11, dadurch gekennzeichnet, daß die verfahrenstechnischen Komponententypen einen hierarchischen Aufbau bilden.

13. Projektierungssystem nach Anspruch 10, dadurch gekennzeichnet, daß die Bibliothek (11) der verfahrenstechnischen Komponententypen eine branchenspezifische Ausprägung hat.

14. Projektierungssystem nach Anspruch 10, dadurch gekennzeichnet, daß die verfahrenstechnischen Komponententypen aus der Sicht eines Technologen definiert sind und ein unmittelbarer Bezug zur Prozeßtechnik besteht.

15. Projektierungssystem nach Anspruch 14, dadurch gekennzeichnet, daß das Verhalten der verfahrenstechnischen Komponententypen hinsichtlich Steuerung und/oder Regelung, Prozeßsimulation und Störungsanalyse spezifiziert ist.

16. Projektierungssystem nach Anspruch 10, dadurch gekennzeichnet, daß für eine konkrete Anlage (12) durch Instanziierung der Verfahrenstechnischen Komponententypen der Anlagenaufbau realisiert ist.

17. Projektierungssystem nach Anspruch 10, dadurch gekennzeichnet, daß die Mittel zur Codegenerierung im Programmspeicher des Rechners geladen sind und automatisiert arbeiten.

18. Projektierungssystem nach Anspruch 17, dadurch gekennzeichnet, daß ein CSL-Compiler (30) vorhanden ist, der zur Codegenerierung genutzt wird.

---

Hierzu 4 Seite(n) Zeichnungen

---

40

45

50

55

60

65

- Leerseite -



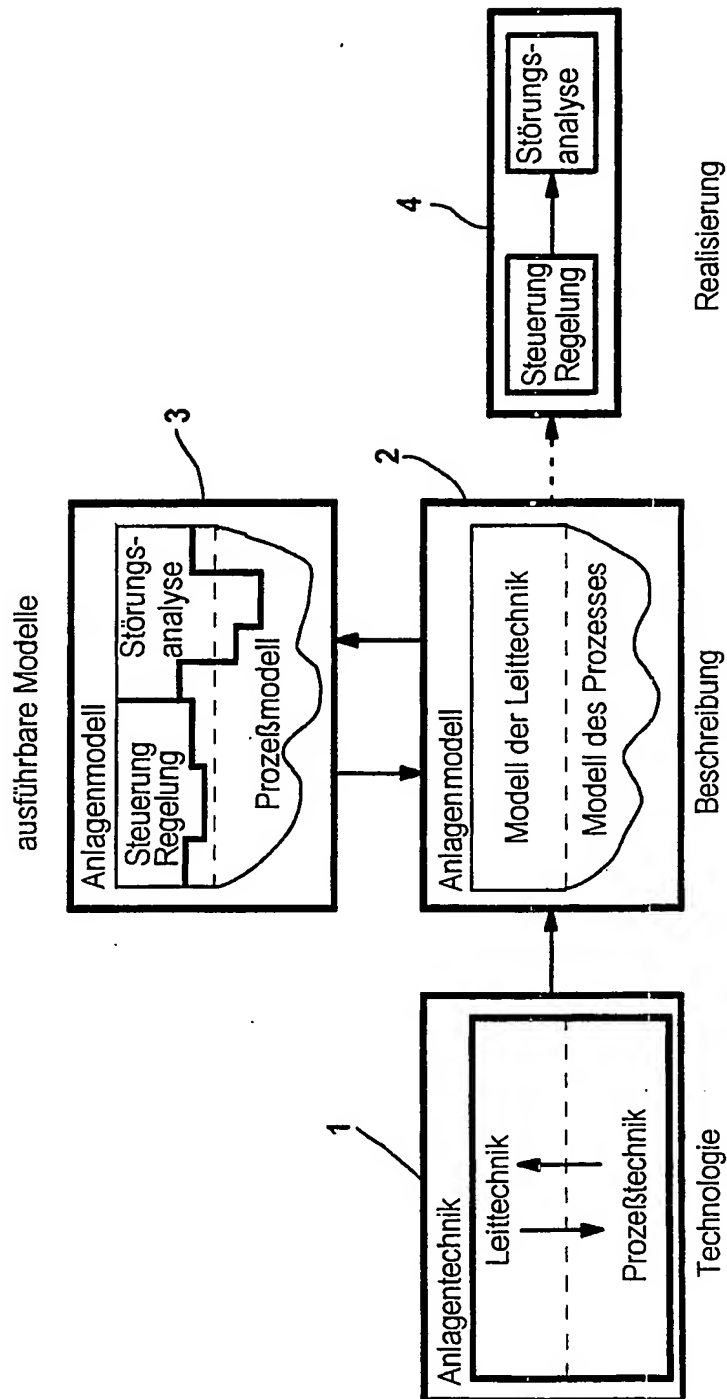


FIG 1

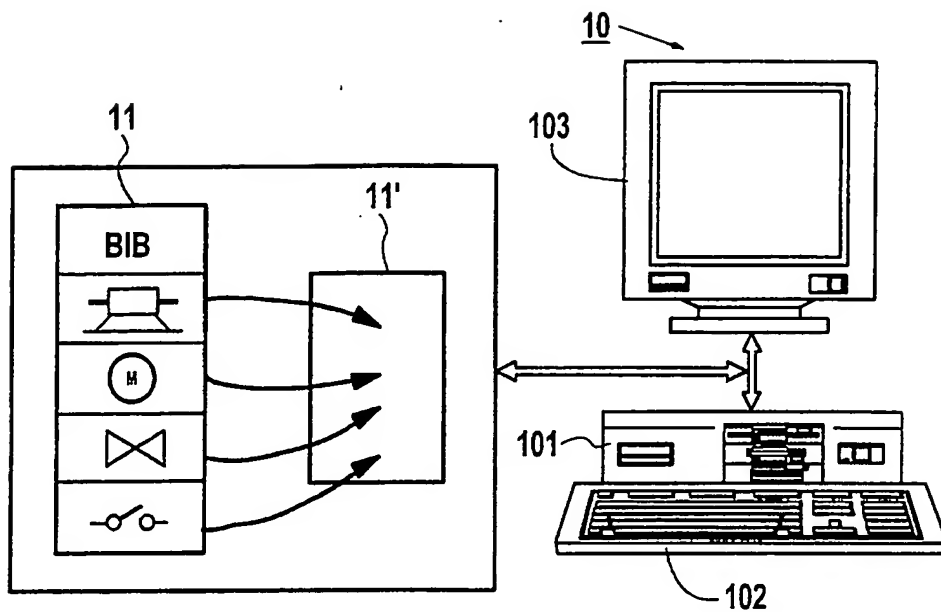


FIG 2

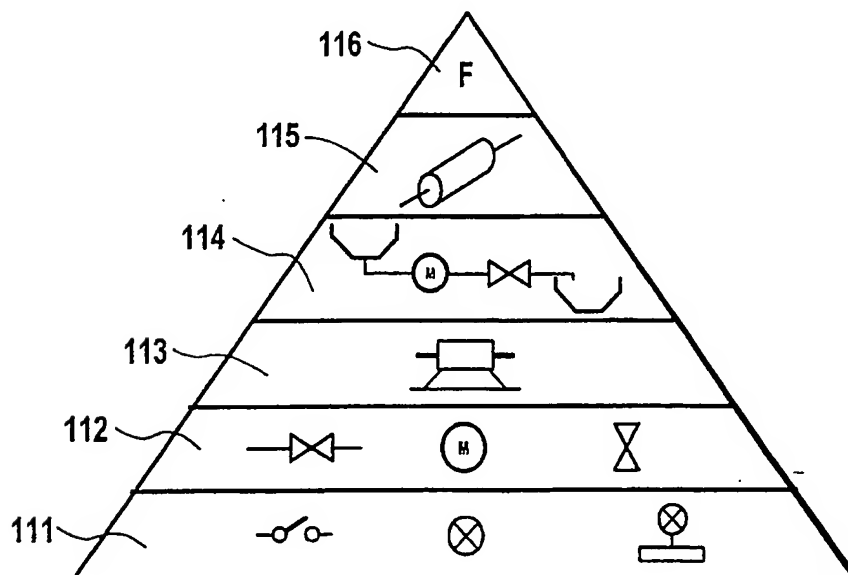


FIG 3

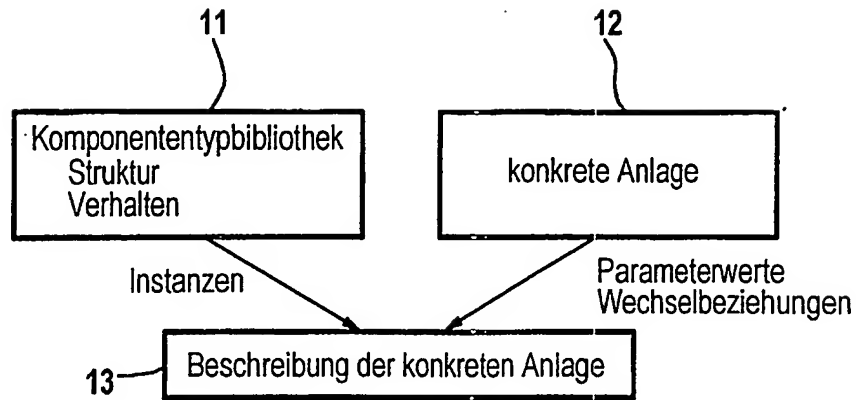


FIG 4

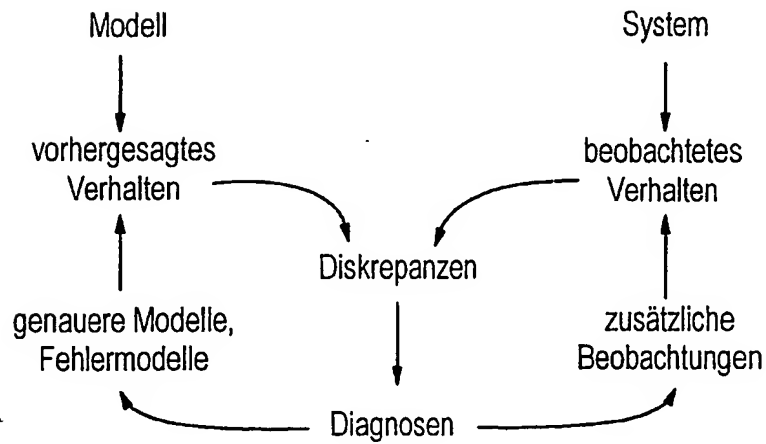


FIG 6

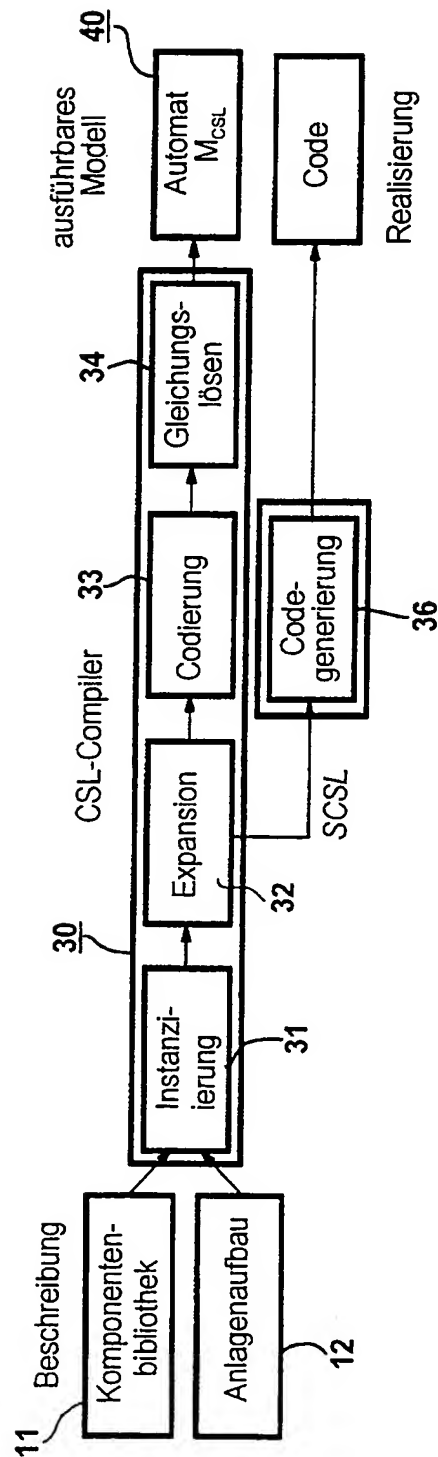


FIG 5